## CLAIMS:

1. (Currently amended) The method of claim 31, further comprising:

sending an encryption request from a first controlled processor in the at least one first controlled processor to the second controlled processor;

receiving, at the second controlled processor, the encryption request;

reading data from the common memory into the local memory associated with the second controlled processor, wherein the reading is performed by the second controlled processor;

executing at the second controlled processor, an encryption process corresponding to the encryption request, the encryption process being adapted to transform the data; and

writing the transformed data from the second controlled processor to the common memory.

2. (Currently amended) The method as described in claim 1 further comprising:

reading, at the second controlled processor, one or more special nonvolatile registers, the special registers including one or more encryption keys; and

using one or more of the encryption keys in the encryption process.

3. (Currently amended) The method as described in claim 1, wherein the sending further comprises writing the request to a mailbox that corresponds to the second controlled processor and the receiving further comprises checking the second controlled processor's mailbox from the second controlled processor.

4. (Original) The method as described in claim 1 further comprising:

identifying an input data area in the common memory from which the data is read and an output buffer area to which the transformed data is written.

5. (Currently amended) The method as described in claim 1, wherein configuring the second controlled processor further comprises:

reading, from the common memory, initialization software code to be executed on the second controlled processor; and

authenticating the initialization software code.

6. (Original) The method as described in claim 5 wherein the authenticating is performed by a routine stored in a nonvolatile memory and wherein the executing of the encryption process is only performed if the initialization software code is successfully authenticated.

7. (Currently amended) The method as described in claim 6 further comprising:

reading, at the second controlled processor, one or more special nonvolatile registers, the special nonvolatile registers including one or more encryption keys, after the initialization software code is successfully authenticated; and

restricting access to the special nonvolatile registers from outside of the second controlled processor.

8. (Currently amended) The method as described in claim 1 wherein the reading and writing steps are performed using Direct Memory Access (DMA) operations.

9. (Currently amended) The method as described in claim 1 further comprising:

identifying the encryption process and an encryption algorithm from a plurality of encryption processes and encryption algorithms based upon the encryption request; and

loading encryption software code corresponding to the identified encryption process and the encryption algorithm, the loading being performed by reading the encryption software code from the common memory to the second controlled processor's local memory.

10. (Canceled)

11. (Currently amended) The information handling system of claim 32, wherein an encryption process runs in the second controlled processor, the encryption process being effective to:

load data, associated with an encryption request, from the common memory to the second controlled processor's local memory;

transform the data based on the encryption request; and

write the transformed data from the second controlled processor's local memory to the common memory.

12. (Currently amended) The information handling system as described in claim 11 further comprising software code effective to:

read, at the second controlled processor, one or more special nonvolatile registers, the special registers including one or more encryption keys; and

use one or more of the encryption keys in the encryption process.

13. (Currently amended) The information handling system as described in claim 11 wherein the encryption request is sent from a first controlled processor in the at least one first controlled processor, and wherein the sending of the encryption request comprises:

writing the encryption request to a mailbox that corresponds to the second controlled processor; and

reading, from the second controlled processor, the encryption request from the second controlled processor's mailbox.

14. (Original) The information handling system as described in claim 11 further comprising software code effective to:

identify an input data area in the common memory from which the data is read and an output buffer area to which the transformed data is written.

15. (Currently amended) The information handling system as described in claim 11 further comprising software code effective to configure the second controlled processor by:

initializing the second controlled processor prior to receiving the request, the initializing further including:

reading, from the common memory, initialization software code to be executed on the second controlled processor; and

authenticating the initialization software code.

16. (Original) The information handling system as described in claim 15 wherein the software code effective to authenticate the initialization software code is performed by a routine stored in a nonvolatile memory, wherein the encryption process is only performed if the initialization software code is successfully authenticated.

17. (Currently amended) The information handling system as described in claim 16 further comprising software code effective to:

read, at the second controlled processor, one or more special nonvolatile registers, the special nonvolatile registers including one or more encryption keys, after the initialization software code is successfully authenticated; and

restrict access to the special nonvolatile registers from outside of the second controlled processor.

18. (Currently amended) The information handling system as described in claim 11 further comprising:

a Direct Memory Access (DMA) controller associated with each of the plurality of controlled processors, wherein the second controlled processor reads from and writes to the common memory using DMA operations performed by the second controlled processor's DMA controller.

19. (Currently amended) The information handling system as described in claim 11 further comprising software code effective to:

identify the encryption process and an encryption algorithm from a plurality of encryption processes and encryption algorithms based upon the encryption request; and load encryption software code corresponding to the identified encryption process and the encryption algorithm, the load being performed by reading the encryption software code [[form]]from the common memory to the second controlled processor's local memory.

20. (Canceled)

21. (Currently amended) The computer program product of claim 33, further comprising:

means for sending an encryption request from a first controlled processor in the at least one first controlled processor to the second controlled processor;

means for receiving, at the second controlled processor, the encryption request;

means for reading data from the common memory into the local memory associated with the second controlled processor, wherein the means for reading is performed by the second controlled processor;

means for executing, at the second controlled processor, an encryption process corresponding to the request, the encryption process being adapted to transform the data; and

means for writing the transformed data from the second controlled processor to the common memory.

22. (Currently amended) The computer program product as described in claim 21 further comprising:

means for reading, at the second controlled processor, one or more special nonvolatile registers, the special registers including one or more encryption keys; and

means for using one or more of the encryption keys in the encryption process.

23. (Currently amended) The computer program product as described in claim 21 wherein the means for sending further comprises means for writing the request to a mailbox that corresponds to the second controlled processor and the means for receiving

further comprises means for checking the second controlled processor's mailbox from the second controlled processor.

24. (Original) The computer program product as described in claim 21 further comprising:

means for identifying an input data area in the common memory from which the data is read and an output buffer area to which the transformed data is written.

25. (Currently amended) The computer program product as described in claim 21, wherein the means for configuring the second controlled processor comprises:

means for initializing the second controlled processor prior to receiving the request, the means for initializing further including:

means for reading, from the common memory, initialization software code to be executed on the second controlled processor; and

means for authenticating the initialization software code.

26. (Previously presented) The computer program product as described in claim 25 wherein the means for authenticating operates using a routine stored in a nonvolatile memory and wherein the means for executing of the encryption process operations only if the initialization software code is successfully authenticated.

27. (Currently amended) The computer program product as described in claim 26 further comprising:

means for reading, at the second controlled processor, one or more special nonvolatile registers, the special nonvolatile registers including one or more encryption keys, the means for reading operating after the initialization software code is successfully authenticated; and

means for restricting access to the special nonvolatile registers from outside of the second controlled processor.

28. (Canceled)

29. (Currently amended) The computer program product as described in claim 21 further comprising:

means for identifying the encryption process and an encryption algorithm from a plurality of encryption processes and encryption algorithms based upon the encryption request; and

means for loading encryption software code corresponding to the identified encryption process and the encryption algorithm, the means for loading operating by reading the encryption software code from the common memory to the second controlled processor's local memory.

30. (Canceled)

31. (Currently amended) A method, in a multiprocessor system, the multiprocessor system comprising a control processor and a plurality of controlled processors, the method comprising:

selecting at least one controlled processor of the plurality of controlled processors to operate in a shared operational state;

selecting a second controlled processor from the plurality of controlled processors to operate in an isolated operational state;

configuring the at least one first controlled processor of the multiprocessor system to be in [[an]] the shared operational state, wherein the shared operational state causes the at least one first controlled processor to operate using a common memory accessible by [[a]] the plurality of controlled processors in the multiprocessor system;

configuring [[a]] the second controlled processor of the multiprocessor system, via loading and executing initialization code in the second controlled processor, to be in [[an]] the isolated operational state, wherein the isolated operational state causes a local memory associated with the [[first]] second controlled processor to be not accessible by the at least one first controlled processor;

executing first code within the [[first]] second controlled processor in a secure manner by virtue of the isolated operational state; and executing second code within the at least one ~~second~~ first controlled processor in an unsecured manner by virtue of the shared operational state.

32. (Currently amended) An information handling system, comprising:

a control processor;

a plurality of controlled processors, wherein each of the plurality of controlled processors comprises a local memory; and

a common memory shared by [[a]] the control processor and the plurality of controlled processors in the information handling system[[;]], wherein the plurality of controlled processors comprises:

at least one first controlled processor~~, in the plurality of processors,~~ selected and configured to be in [[an]] a shared operational state, wherein the shared operation state causes the at least one first controlled processor to operate using the common memory ~~accessible by a plurality of processors in the information handling system~~; and

a second controlled processor selected and configured, via loading and executing initialization code in the second controlled processor, to be in an isolated operational state, wherein the isolated operational state causes a local memory associated with the [[first]] second controlled processor to be not accessible by the at least one first controlled processor, wherein the [[first]] second controlled processor executes first code in a secure manner by virtue of the isolated operational state, and wherein the at least one ~~second~~ first controlled processor executes in an unsecured manner by virtue of the shared operational state.

33. (Currently amended) A computer program product comprising a computer useable medium having a computer readable program, wherein the computer readable program, when executed on a computing device comprising a control processor and a plurality of controlled processors, causes the computing device to:

select at least one first controlled processor of the plurality of controlled processors to operate in a shared operational state;

select a second controlled processor from the plurality of controlled processors to operate in an isolated operational state;

configure the at least one first controlled processor of the computing device to be in [[an]] the shared operational state, wherein the shared operational state causes the at least one first controlled processor to operate using a common memory accessible by [[a]] the plurality of controlled processors in the computing device;

configure [[a]] the second controlled processor of the computing device, via loading and executing initialization code in the second controlled processor, to be in [[an]] the isolated operational state, wherein the isolated operational state causes a local memory associated with the [[first]] second controlled processor to be not accessible by the at least one first controlled processor;

execute first code within the [[first]] second controlled processor in a secure manner by virtue of the isolated operational state; and

execute second code within the at least one ~~second~~ first controlled processor in an unsecured manner by virtue of the shared operational state.

34. (New) The method of claim 31, wherein selecting a second controlled processor from the plurality of controlled processors to operate in an isolated operational state comprises:

identifying a free controlled processor in the plurality of controlled processors that has not be dedicated to perform a specific device function; and

assigning the free controlled processor to be the second controlled processor and perform encryption device functions.

35. (New) The method of claim 31, wherein configuring the second controlled processor to be in the isolated operational state comprises:

determining if the initialization code is authentic;

setting the second controlled processor to run in the isolated operational state in response to the initialization code being determined to be authentic; and providing access to special purpose registers storing encryption keys to only the second controlled processor in response to the setting of the second controlled processor to run in the isolated operational state.

36. (New) The method of claim 35, wherein if the initialization code is determined to not be authentic, the second controlled processor is set to run in the shared operational state and another controlled processor in the plurality of controlled processors is selected to operate in the isolated operational state.

37. (New) The method of claim 31, wherein the common memory comprises a first portion associated with the control processor and a second portion associated with the plurality of controlled processors, and wherein data to be processed by the second controlled processor in the isolated operational state is retrieved from the first portion of the common memory and results data generated by processing the data is written back to the first portion of the common memory.